

**REMARKS**

New dependent claims 25 - 30 are added. Hence, claims 1 - 30 are all the claims pending in the application. Claims 1, 4, 9, 12, 17 and 20 are amended merely to correct informalities and to recite the invention more clearly. The claims are not amended in a manner that narrows the scope of the claims nor are they amended for reasons related to patentability and it is respectfully submitted that the claims would be allowable without the amendments.

Claims 1-24 are rejected under 35 USC §102(e) as being anticipated by Kauffman et al. ("Kauffman"). Applicant respectfully traverses the rejection because Kauffman et al. does not disclose all the limitations of the claims.

Claim 1 is directed to a method for adding user-provided content to a content object that is stored as a plurality of content entities in a data repository. The method includes defining the content object by a list of content entity identifiers and receiving user-provided content and assigning it an identifier. The user-provided content and its identifier are stored in the data repository. The identifier of the user-provided content is added to the list of content entity identifiers, thereby adding the user-provided content to the content object.

Kauffman relates to methods for handling large digital objects in a client/server library system. Those large objects are handled by dividing them into smaller pieces which are stored in the digital library system. See Abstract. Kauffman discloses that when an application program stores a large object in the digital library, a library client 30, shown in Fig. 2, divides it into individual pieces. See col. 9, lines 58-59. The library client creates a piece map 26 that points to the individual pieces (col. 7, line 56) and sends a request to a library server 20 to store the piece map in the library. As shown in Fig. 4, the library server creates an entry in its parts table 14 for the piece map and assigns it an ID and a part number. See col. 9, line 64 et seq. As each piece

of the large object is stored in the library, the piece map is updated to include the piece number of that piece of the large object. Once all the pieces of the large object are stored, the library server 10 sends a response message back to the library client 30 indicating that the last piece of the large object has been stored. The library client then notifies the application program that the object has been stored. See col. 10, lines 47-57.

The Office Action states that Kauffman, at col. 9, line 64 through col. 10, line 13, discloses adding in item ID, part number and rep type to a list, and asserts that this corresponds to adding client-provided content to an object. That cited portion of Kauffman describes a large object that has been divided into individual pieces being stored in the digital library and recording information concerning those individual pieces in a piece map. Kauffman discloses that the piece map is assigned an item ID, a part number, and a rep type which are recorded in a parts table 14 shown in Fig. 4. The Examiner's position appears to be that the list referred to at the top of page 3 in the Office Action, corresponds to Kauffman's parts table 14, and the information concerning the piece map, namely, the item ID, part number and rep type, are added to the parts table. However, the item ID, part number and rep type are not user-provided content as recited in claim 1. Hence, Kauffman does not meet all the limitations of the claim.

Even if the Examiner's position is that Kauffman's large object corresponds to the content object recited in claim 1, that the piece map corresponds to the claimed list, and that the pieces of the large object correspond to the claimed user-provided content, Kauffman still does not teach or suggest all the limitations of claim 1. Claim 1 recites receiving user-provided content, assigning it an identifier, and adding that identifier of the user-provided content to a list of content entity identifiers that define the content object, whereby the user-provided content is

added to the content object. Kauffman, however, does not disclose receiving user-provided content and adding an identifier of it to the piece map such that the user-provided content is added to the large object. Rather, Kauffman discloses dividing a large object and storing its pieces, which are already part of the large object, in a digital library and using the piece map to keep track of those pieces. Accordingly, it is respectfully submitted that Kauffman does not anticipate claim 1.

The remaining independent claims, namely claims 4, 9, 12, 17 and 20, each recite receiving user-provided content and adding an identifier of that user-provided content to an outline or list to thereby add the user-provided content to the content object. Accordingly, it is respectfully submitted that those claims are not anticipated by Kauffman for at least the same reasons discussed above. Since all the remaining claims incorporate by reference all the limitations of one of the independent claims discussed above, it is respectfully submitted that those dependent claims are patentable for at least the same reasons.

New dependent claims 25-30 are added and recite that the user-provided content that is received is not part of the content object. It is respectfully submitted that these new dependent claims are patentable for at least the same reasons as the claims from which they depend.


In view of the foregoing, Applicant respectfully requests the Examiner to find the application in condition for allowance. However, if for any reason the Examiner believes that the application is not now in condition for allowance, the Examiner is respectfully requested to call the undersigned to resolve any issues and to expedite the disposition of the application.

**Amendment**

**U.S. Patent Appln. No. 09/488,976**

Applicant hereby petitions for any extension of time that may be required to maintain the pendency of this case, and any required fee for such extension is to be charged to Deposit Account No. 05-0460.

Respectfully submitted,



J. Warren Lytle, Jr.

Registration No. 39,283

EPSTEIN, EDELL, SHAPIRO, FINNAN & LYTLE, LLC  
1901 Research Blvd., Suite 400  
Rockville, Maryland 20850-3164  
(301) 424-3640

Hand Delivered on: July 3, 2002

**Version With Markings to Show Changes Made**

Set forth below are the replacement paragraphs of the specification rewritten in the accompanying Amendment, marked up to show all changes relative to the previous version of those claims, in accordance with 37 C.F.R. §1.121(b)(1)(iii).

Amend the specification beginning at page 1, line 10 and ending at page 1, line 50, as follows.

**A System and Method for Creating Compilations of Content**

Serial No. [ ]/[ ], [ ] 09/489,134 (Our reference Docket # STL000012US1) *Pham Hung* (TV)

**Method and System for Adding Content to a Content Object Stored in a Data Repository**

Serial No. [ ]/[ ], [ ] 09/489,576 (Our reference Docket # STL000013US1) *Herndon, Heather* (TV)

**Method and System for Moving Content in a Content Object Stored in a Data Repository**

Serial No. [ ]/[ ], [ ] 09/488,971 (Our reference Docket # STL000015US1) *Herndon, Heather* (TV)

**Method and System for Removing Content in a Content Object Stored in a Data Repository**

Serial No. [ ]/[ ], [ ] 09/489,087 (Our reference Docket # STL000016US1) *CHANNARAJALA, SARANA* (TV)

**Prerequisite Checking in a System for Creating Compilations of Content**

Serial No. [ ]/[ ], [ ] 09/488,969 (Our reference Docket # STL000017US1) *Herndon, Heather* (TV)

**Method and System for Preventing Mutually Exclusive Content Entities Stored in a Data Repository to be Included in the Same Compilation of Content**

Serial No. [ ]/[ ], [ ] 09/489,265 (Our reference Docket # STL000018US1) *Truong Lam* (TV)

**Volume Management Method and System for a Compilation of Content**

Serial No. [6/444,627] 09/489,090 (Our reference Docket # STL000019US1) *Pham Hung* (TV)

**Method and System for Calculating Cost of a Compilation of Content**

Serial No. [ ]/[ ], [ ] 09/489,143 (Our reference Docket # STL000020US1) *Herndon, Heather* (TV)

**Method and System for Storing Hierarchical Content Objects in a Data Repository**

Serial No. [ ]/[ ], [ ] 09/489,570 (Our reference Docket # STL000021US1) *Pham Hung* (TV)

**File Structure for Storing Content Objects in a Data Repository**

Serial No. [ ]/[ ], [ ] 09/489,730 (Our reference Docket # STL000022US1) *Pham Hung* (TV)

**Amendment**

**U.S. Patent Appln. No. 09/488,976**

**Providing a Functional Layer for Facilitating Creation and Manipulation of Compilations of Content** <sup>fin</sup>

Serial No. [ ] / [ ], [ ] <sup>WOMEN, LAM NTH</sup> 09/489,605 (Our reference Docket # STL000023US1) (TV)

**A Hitmask for Querying Hierarchically Related Content Entities** <sup>WOOD, ISAAC</sup>

Serial No. [ ] / [ ], [ ] 09/489,133 (Our reference Docket # STL990182US1) (TV)

**A Method and Configurable Model for Storing Hierarchical Data in a Non-Hierarchical Data Repository** <sup>J. K. TRI, ANIL</sup>

Serial No. [ ] / [ ], [ ] 09/489,561 (Our reference Docket # STL000025US1) (TV)

**Reference to a Computer Listing Appendix**

Appendix A to this application is set forth on a single compact disc and the material recorded thereon is incorporated by reference herein. The following file is recorded on the compact disc: file name: AppendixA.txt; file size: 107kB; date of creation: May 16, 2002.--

Amend the paragraph beginning at page 6, line 3, as follows.

Figs. 22A – [22D] 22E represent the system administrator interface of an embodiment of the present invention;

Amend the paragraph beginning at page 6, line 7, as follows.

Fig. [25] 24 is a state diagram representing the states of a user, request and CBO at various stages of the process for creating compilations of content.

**Claims**

Set forth below are the claims rewritten in the accompanying Amendment, marked up to show all changes relative to the previous version of those claims, in accordance with 37 C.F.R. §1.121(c)(ii).

1. (Amended) A method for adding user-provided content to a content object stored as a plurality of content entities in a data repository, comprising the steps of:

[Defining] defining the content object by a list of content entity identifiers;

[Receiving] receiving user-provided content, assigning it an identifier, and storing it with

5 its identifier in the data repository; and

[Adding] adding the identifier of the user-provided content to the list, whereby the user-provided content is added to the content object.

4. (Amended) A method for adding user-provided content to a hierarchically structured content object stored as a plurality of content entities in a data repository, comprising the steps of:

[Defining] defining the content object by a hierarchical outline of containers and content

5 entity identifiers;

[Receiving] receiving user-provided content, assigning it an identifier, and storing it with its identifier in the data repository; and

[Adding] adding the identifier of the user-provided content to the outline, thereby adding the user-provided content to the content object.

9. (Amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for adding user-provided content to a content object stored as a plurality of content entities in a data repository, comprising the steps of:

5 [Defining] defining the object by a list of content entity identifiers;

[Receiving] receiving user-provided content, assigning it an identifier, and storing it with its identifier in the data repository; and

[Adding] adding the identifier to the list, whereby the user-provided content is added to the content object.

12. (Amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for adding user-provided content to a hierarchically structured content object stored as a plurality of content entities in a data repository, comprising the steps of:

5 [Defining] defining the content object by a hierarchical outline of containers and content entity identifiers;

[Receiving] receiving user-provided content, assigning it an identifier, and storing it with its identifier in the data repository; and

10 [Adding] adding the identifier to the outline, thereby adding the user-provided content to the content object.

17. (Amended) A system for adding user-provided content to a content object stored as a plurality of content entities in a data repository, comprising:

[Means] means for defining the object by a list of content entity identifiers;

5 [Means] means for receiving user-provided content, assigning it an identifier, and storing it with its identifier in the data repository; and

[Means] means for adding the identifier to the list, whereby the user-provided content is added to the content object.



20. (Amended) A system for adding user-provided content to a hierarchically structured content object stored as a plurality of content entities in a data repository, comprising the steps of:

[Means] means for defining the content object by a hierarchical outline of containers and  
5 content entity identifiers;

[Means] means for receiving user-provided content, assigning it an identifier, and storing it with its identifier in the data repository; and

[Means] means for adding the identifier to the outline, thereby adding the user-provided content to the content object.